A Comparison of Classifiers for Intelligent Machine Usage Prediction

Chiming Chang, Paul-Armand Verhaegen, Joost R. Duflou
Centre for Industrial Management,
Department of Mechanical Engineering,
KU Leuven, Belgium
{Chiming.Chang, PaulArmand.Verhaegen}@cib.kuleuven.be, Joost.Duflou@mech.kuleuven.be

***Abstract*—Probability estimation of machine usages is an essential task to the development of an intelligent device/environment. In this paper, we propose a generic framework to the task using the sliding window technique and incremental feature selection. The methodology is applied to a real-life dataset of office printers and the performances of different standard classifiers in supervised learning are compared. We conclude that Logistic Regression (LR) outperform other classifiers and is appropriate for the proposed framework. The use of Generic Bayesian Network (GBN) classifier is also promising, if combined with feature reduction methods.**

I. Introduction

Probability estimation of machine usage is an important and essential task for intelligent device/environment design. We envisage an environment where one or multiple machines are installed, and the usage of machines are logged and stored. The goal is to build a predictive model from the stored usage history which reflects the usage pattern and user behaviors. Examples are predicting the usage of domestic appliances in a smart house, or predicting the usage for an independent intelligent machine, such as a coffee-maker which saves energy based on the usage prediction. The probability estimations from the model can be used by a system controller either to adjust the machine behavior or to provide useful suggestions to machine users.

The probability estimation of machine usage fits into the problem of *probability forecasting for categorical time series*. One common approach is to transform the problem into a standard supervised learning problem by sliding window technique [1]. The obvious advantage to this approach is that different standard classifiers in supervised learning can be applied to the problem once it is transformed. Also, it provides a simple solution to handle the multiple time series scenario when environmental factors or inter-machine dependencies are considered.

In [2], prediction accuracy of standard classifiers (decision table, decision tree and Bayes networks) for appliance usage prediction are compared; however, the aspect of probability estimation is less stressed. Appliance usage modeling with graphical models and Dynamic Bayesian Network have also been examined in previous researches([3], [4]). The major difficulty of these methods is that it depends on good domain knowledge to define a proper network structure so the solution is often case specific.

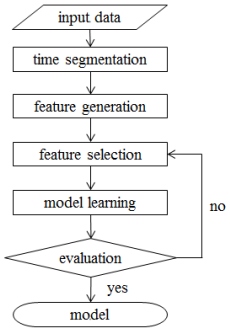

Fig. 1. Generic framework for machine usage profiling.

We explain the generic framework of building a usage model from historical data in section II. In section III, we implement the framework to a dataset of office printers, in order to evaluate the capabilities of standard classification algorithms of usage prediction task.

II. Framework for Usage Modeling

The proposed framework for machine usage modeling is illustrated as Figure 1. The main idea is to generate several attribute sets and select the relevant attribute sets incrementally by a wrapper approach. We detailed each phase of the framework below.

1) input data: Input data typically consist of one or multiple time series, and there is a main time series which represents the target machine usage to be modeled. Other time series represent usages of peer machines and environmental variables and some of them may be correlated to the main series and provide additional information for prediction. The exact correlations between time series, however, are case dependent and have to be found out with algorithms.

For each sample in a time series, corresponding timestamps are specified. Input data typically come from various sensors or measuring equipment and take the form of $\langle$*timestamp, measurement*$\rangle$ 2-tuples. As an example, the data may consist of two time series representing the power measurement of the machine and the presence of its users.

2) time segmentation: To beginning with, a usage history to be modeled is selected and is segmented into evenly spaced time slots. The size of time slots is selected according to the

need of system controller. Typical values may be a quarter or an hour. Input data are converted into a multivariate time series given the segmentation and can be generalized below.

$$\begin{array}{llll} t_1, & t_2, & \ldots, & t_N \\ y_1, & y_2, & \ldots, & y_N \\ x_{1,1}, & x_{1,2}, & \ldots, & x_{1,N} \\ x_{2,1}, & x_{2,2}, & \ldots, & x_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m,1}, & x_{m,2}, & \ldots, & x_{m,N} \end{array}$$

In the segmented time series, N is the total number of time slots to be modeled. Sequence T indicates the timestamps of each time slot. Sequence Y is a nominal variable which represent the usage of machine for each time slot. A predefined rule transforms the real value measurement of input data into nominal values, such as machine usage $\langle$*used, not-used*$\rangle$ or machine status $\langle$*on, stand-by, off*$\rangle$ depending on the application. Variables X_1 to X_m represent environmental attributes that are measured concurrently and are possible (or not) to provide extra information for prediction. Measurements of environmental sensors are also segmented and labeled for each time slot. For example, $x_{1,i} \in \{0, 1\}$ represents the presence of users and $x_{2,i} \in \{cold, warm, hot\}$ represents outdoor temperature at time slot i.

3) feature generation: Standard supervised learning algorithms assume that instances are sampled from an i.i.d distribution. For time series data, the time domain correlation can be represented using the sliding window technique[5].

The segmented time series is considered a classification problem of N samples by assigning Y for class variable. Different attributes are generated and arranged as different attribute sets. We describe these attribute sets below.

1) temporal attributes:
   Attributes representing temporal information are generated from timestamps T. Some attributes are trivial, such as days of the week and hours of the day. Some are more intricate and depending on the domain knowledge of the system, such as holidays and/or seasons.
2) historical usage attributes:
   Historical usage attributes represent the pattern and dynamic of usage behaviors. A sliding window with predefined size w is applied to select $\langle y_{i-1}, y_{i-2}, \ldots y_{i-w} \rangle$ as attributes to predict class value y_i.
3) concurrent environmental attributes:
   Labels of environment attributes at target time slot i, i.e. $(x_{1,i}, x_{2,i}, \ldots x_{3,i})$ are the third type of attributes. These attributes indicate the effect of the environmental situation to machine usage.
4) historical environmental attributes:
   For each time sequence of an environmental attribute, a predefined window size is given and data in the window are selected as a separate attribute set. In total, m attribute sets are generated for m environmental attributes.

To learn the machine usage model with all generated attributes are infeasible and usually leads to lousy results, since only a part of the generated attributes contain critical information for usage prediction and others are less correlated or too noisy. A feature selection strategy is therefore necessary to select critical features and to optimize the model.

4) feature selection: According to the experience, temporal attributes are the most informative in usage prediction and the importance of other attribute sets are usually case dependent. An incremental feature selection using the wrapper method is applied to find the combination of attribute sets that maximizes the performance scores of probability estimation. Firstly, a preliminary predictive model is learned with only temporal attributes. A heuristic search algorithm, e.g. hill climbing, then optimizes the model by adding attribute sets which improve performance scores.

5) model learning: Although, all classification algorithms that produce probability estimations can be used for model learning. The expected classification algorithm should provide good probability estimations, avoid over-fitting and can be executed in reasonable time. In this work, we compare the performance of standard classification algorithms for usage modeling and check their feasibility for the proposed framework.

6) evaluation: The evaluation phase checks performance scores of the learned model and controls the iterations of heuristic search. The search algorithm may stop when all attribute sets are evaluated, a predefine accuracy level is satisfied or the modeling time exceeds a predefined limit.

Quality measures of probability estimations have been widely studied in previous researches([6], [7]). We examine popular quality measures including logarithm loss, squared error loss, and ROC curve in this work. Cross validation is applied to avoid over-fitting.

III. Experiment

We conduct an experiment to compare the performance of standard classifiers for probability estimations in machine usage modeling. By examining model quality, computational complexity and training time, we try to conclude which classifiers are appropriate for the proposed framework.

A. Dataset

The dataset used is a data log of printer usage in an office building of KU Leuven. The data log consists of the usage of 53 printers over a 3 years period, from 2009 to 2012. However, some of the printers have only a short usage history and some are used sparsely. After removing these printers, 21 printers were left for analysis.

For each printer, a usage history of 40 weeks is arbitrarily chosen and the time slot of one hour is selected. Machine usage is assigned to be 1 if there is any printing request in the time slot, or 0 otherwise.

The usage history of 40 weeks is selected mainly because it reflects the real situation where the system is expect to monitor the machine usage for a few months and apply control polices as soon as reliable usage patterns are detected. Besides, some smaller simulations show that different sizes of time slot

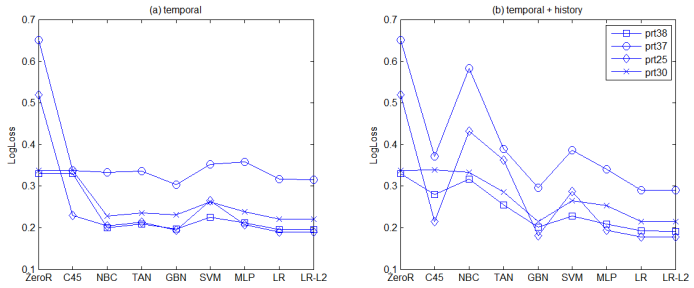

Fig. 2. Comparison of LogLoss for 4 distinct printers modeled with different classifiers.

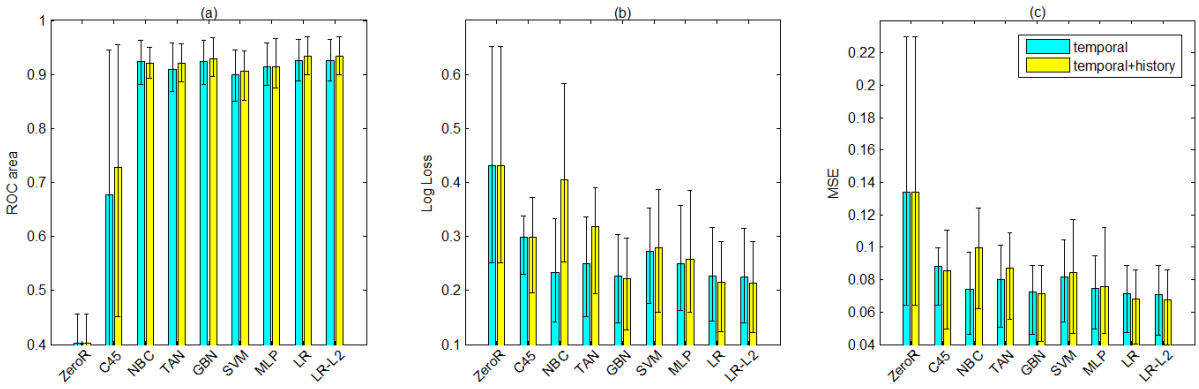

Fig. 3. Averaged performance metrics of classifiers for all tested printers with error bars.

(20 minutes to 2 hours) does not effect the usage prediction significantly.

B. Experiment Design

Several standard classification algorithms provided by the popular machine learning tool WEKA[8] are applied to learn the usage model for each printer, i.e. the classification model for $y_i \in \{0, 1\}$. The classifiers tested are Decision Tree (C4.5), Naive Bayesian Classifier (NBC), Tree Augmented Naive Bayes (TAN), General Bayesian Network (GBN)[9], Multilayer Perceptron (MLP), Logistic Regression (LR) and Support Vector Machine (SVM). For each algorithm, a tuned parameter setting is applied to all printers. Parameters are not optimized for individual printer.

Two scenarios are tested. In the first scenario, only temporal attributes are used for usage modeling. Temporal attributes are generated from time-stamp of individual time slot, including hours of the day (HR), days of the week (WD), the month (MON) and public holidays (HOL). This is to test how different classifiers estimate the conditional probability of $P(Y|HR, WD, MON, HOL)$, since we consider that temporal attributes provide the fundamental information content for machine usage prediction.

In the second scenario, the usage models consider both temporal attributes and the printer usage of previous 12 hours (a sliding window of 12 hours). This scenario simulates the iteration of heuristic search, i.e. adding auxiliary attribute sets to improve the precision of probability estimation.

C. Results

The results of experiments are illustrated as Figure 2 and 3. All performance metrics are calculated based on the probability estimations of a 20-fold cross validation to avoid over-fitting. In Figure 2, the logarithm losses (LogLoss) of four distinct printers are compared; it shows how different classifiers perform on individual printers. Even though the predictabilities of printer usages vary with printers, due to different user behaviors and randomness of data, the performances of algorithms are consistent between different printer data.

Figure 3 compares several performance metrics for all printers under test, including the area under receiver operating characteristic (ROC) curve, logarithm loss (LogLoss), mean squared error loss (MSE). The means of performance metrics are illustrated with error ranges labeled. The results of both scenarios are shown together for each classifier to facilitate comparison.

The ZeroR classifier is listed for baseline comparison. It gives probability forecasting based on empirical distribution of Y to all instances indiscriminately. The C4.5 decision tree can model some of the printer data but for some printers the performance downgrade to that of ZeroR (prt38 and prt30 in Figure 2). This happens when the usage of printers is too sparse or too random and all branches are removed after pruning. This can be fixed by adjusting pruning parameters, but this is normally data dependent. Also, the performance of decision tree is worse than other classifiers even when the

decision tree can be reasonably built.

Three types of Bayesian network classifiers are evaluated, including Naive Bayesian Classifier (NBC), Tree Augmented Naive Bayes (TAN), and General Bayesian Network (GBN). The GBN algorithm used in our experiment is adapted from standard Weka API. It starts with a Naive Bayes structure and search an optimal graph that maximizes Conditional Log Likelihood (CLL) by adding or removing arcs. The nodes of Bayesian Network are assigned with empirical conditional probabilities.

NBC and TAN work proper while modeling with only temporal attributes; in such case the number of attributes is limited and almost all attributes are significant. Both of the two are, however, subject to a fixed graphical structure and lack the ability to disregard irrelevant attributes. The prediction error increases for prt37 and prt25 in Figure 2(b) when many irrelevant attributes (long distance historical usage) are considered in the model.

The GBN, on the contrary, can compete with top ranking classifiers, such as LR, by searching an adequate Bayesian network which excludes irrelevant attributes. However, the computing time for optimal network searching increases exponentially with the number of nodes (Table I). In fact, in our experiment, the selected window size for GBN is 4 hours rather than 12 hours.

In general, we expect the prediction models which combine temporal attributes and historical usages to perform better than the ones using only temporal attributes, but this would require the algorithm to have some embedded feature selection ability to include only relevant attributes of the usage history to the model. This is especially important to the proposed framework, since in each iteration it evaluates an attribute set which may contain both relevant and irrelevant attributes.

MLP and SVM algorithms are time consuming but the performance of prediction is less satisfying. The modeling time is not a critical concern since in our application the machines are expected to be modeled every few days and the modeling time increases only linearly with the number of attributes. Both MLP and SVM can attain high classification accuracy by modeling non-liner decision boundary for class labels. However, for small dataset cases, such as usage modeling, they tend to over-fitting the training sets and perform poorly on test sets.

LR algorithms outperform other classifiers in the experiment. The modeling time is short and the parameters are easy to tune. The ridge regression in LR algorithms provides embedded feature selection ability. Two LR algorithms are test for validation, one from standard Weka API and one from LIBLINEAR library [10]. Both algorithms show similar results.

Overall, the results in Figure 3 show that LR and GBN perform better than others, and can effectively exploiting the useful information in auxiliary datasets (in this case usage history). We conclude that it is therefore appropriate to use LR as the core classification algorithm for developing more complicate modeling strategies, such as the framework proposed. The use of GBN is also promising if combined with additional feature reduction methods.

TABLE I
AVERAGED MODEL LEARNING TIME FOR ALL CLASSIFIERS IN BOTH SCENARIOS

Classifier	C45	NBC	TAN	GBN	SVM	MLP	LR
temporal	3.0	2.8	3.1	3.8	579.0	380.9	36.8
temporal+history	3.9	2.9	5.9	31.2*	781.3	1330.0	54.5

Note: GBN uses 4-hour usage history instead of 12-hour.

IV. Conclusion and Future Work

We proposed a generic framework for machine usage modeling which can be used in intelligent device design or as knowledge base for smart environment agents. The proposed framework converts the time series prediction of machine usage into standard supervised learning problems using the sliding window method and learns the predictive model by searching a combination of attributes sets which maximizes performance scores. The experiment shows that Logistic Regression (LR) outperforms other standard supervised learning algorithms while modeling machine usages under the proposed framework.

The framework and the experiment are preliminary results in our research of machine usage modeling. Some further research topics include automatic sliding window size selection, adding feature extraction steps to improve probability estimation performance, and to implement practical GBN modeling by combining feature reduction steps.

References

[1] T. G. Dietterich, "Machine learning for sequential data: A review," in *Structural, Syntactic, and Statistical Pattern Recognition*. Springer-Verlag, 2002, pp. 15–30.

[2] K. Basu, L. Hawarah, N. Arghira, H. Joumaa, and S. Ploix, "A prediction system for home appliance usage," *Energy and Buildings*, vol. 67, no. 0, pp. 668 – 679, 2013.

[3] N. C. Truong, J. McInerney, L. Tran-Thanh, E. Costanza, and S. D. Ramchurn, "Forecasting multi-appliance usage for smart home energy management," in *23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*, April 2013.

[4] Z.-H. Lin and L.-C. Fu, "Multi-user preference model and service provision in a smart home environment," in *Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on*, Sept 2007, pp. 759–764.

[5] D. Lindsay and S. Cox, "Effective probability forecasting for time series data using standard machine learning techniques," in *Pattern Recognition and Data Mining*, 2005, pp. 35–44.

[6] V. Vapnik, "An overview of statistical learning theory," *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 988–999, Sep 1999.

[7] C. E. Metz, "Basic principles of roc analysis," *Seminars in Nuclear Medicine*, vol. 8, no. 4, pp. 283 – 298, 1978.

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[9] J. Cheng and R. Greiner, "Comparing bayesian network classifiers," in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 101–108.

[10] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear - a library for large linear classification," 2008, the Weka classifier works with version 1.33 of LIBLINEAR. [Online]. Available: http://www.csie.ntu.edu.tw/čjlin/liblinear/